Preprint
UCRL-JC-135010

Adaptive Node Techniques for Maxwell's Equation

D.W. Hewett

This article was submitted to
Numerical Simulations of Plasmas,
Banff, Canada, May 22, 2000

U.S. Department of Energy
Lawrence
Livermore
National
Laboratory

April 1, 2000

Adaptive Node Techniques for Maxwell's Equations

D.W. Hewett, LLNL

Introduction

The computational mesh in numerical simulation provides a framework on which to monitor the spatial dependence of function and their derivatives. Spatial mesh is therefore essential to the ability to integrate systems in time without loss of fidelity. Several philosophies have emerged to provide such fidelity (Eulerian, Lagrangian, Arbitrary Lagrangian Eulerian ALE, Adaptive Mesh Refinement AMR, and adaptive node generation/deletion). Regardless of the type of mesh, a major difficulty is in setting up the initial mesh. Clearly a high density of grid points is essential in regions of high geometric complexity and/or regions of intense, energetic activity. For some problems, mesh generation is such a crucial part of the problem that it can take as much computational effort as the run itself, and these tasks are now taking weeks of massively parallel CPU time.

Mesh generation is no less crucial to electromagnetic calculations. In fact EM problem set up can be even more challenging without the clues given by fluid motion in hydrodynamic systems. When the mesh is advected with the fluid (Lagrangian), mesh points naturally congregate in regions of high activity. Similarly in AMR algorithms, strong gradients in the fluid flow are one of the triggers for mesh refinement.

In the hyperbolic Maxwell's equations without advection, mesh point placement/motion is not so intuitive. In fixed geometry systems, it at least feasible to finely mesh high leverage, geometrically challenged areas. For other systems, where the "action" takes place far from the boundaries and, likely, changes position in time, the options are limited to either using a high resolution (expensive) mesh in all regions that could require such resolution or adaptively generating nodes to resolve the physics as it evolves. I have developed a new time of adaptive node technique for Maxwell's equations to deal with this set of issues.

Finite-Differencing of Maxwell's Potential Equations

In the Lorentz gauge, the wave equation for both the scalar and vector potentials can be integrated in time using a leapfrog technique for each component of potential and its time-derivative. The wave equations are

$$\frac{\partial^2\Phi}{\partial t^2} - c^2\,\nabla^2\Phi = 4\pi c^2\,\rho \qquad \frac{\partial^2 \vec{A}}{\partial t^2} - c^2\,\nabla^2 \vec{A} = 4\pi\, c\, \vec{J}$$

with Lorentz condition

$$c\,\nabla\cdot\vec{A} + \frac{\partial\Phi}{\partial t} = 0 \quad .$$

Defining $\phi = \Phi$ and $\vec{a} = \dot{\vec{A}}$, we can finite difference the wave equations

$$\phi_{n+1/2} = \phi_{n-1/2} + c^2 \Delta t \nabla^2 \Phi_n + 4\pi c^2 \Delta t \rho_n \qquad \vec{A}_{n+1/2} = \vec{A}_n + \frac{\Delta t}{2} \vec{a}_n$$

$$\Phi_{n+1} = \Phi_n + \Delta t \phi_{n+1/2} \qquad \vec{a}_{n+1} = \vec{a}_n + c^2 \Delta t \nabla^2 \vec{A}_{n+1/2} + 4\pi c \Delta t \vec{J}_{n+1/2}$$

$$\vec{A}_{n+1} = \vec{A}_{n+1/2} + \frac{\Delta t}{2} \vec{a}_{n+1}$$

so that the required electric and magnetic fields at the advanced time can be recovered from

$$\vec{E}_{n+1} = -\nabla \Phi_{n+1} - \vec{a}_{n+1} / c \qquad \vec{B}_{n+1} = \nabla \times \vec{A}_{n+1} \ .$$

For plasma physicists, these equations are more unusual than profound. The motivation for the choice of Lorentz gauge is so that the both/either the scalar and magnetic vector potential, Φ and $\vec{A}$, can be independently integrated in time. Like the more traditional Yee algorithm, it is straightforward to evaluate the Laplacians and thus advance these equations locally if the nearby neighbors can be identified. Stable time integration of these equations, however, requires a few more tricks.

A Virtual Spatial Array of Node Locations

A spatial array of nodes, where values of the potentials and their time derivatives are stored, forms a "mesh" on which we can advance the electromagnetic field. Although the resulting explicit scheme is still subject to CFL, the appeal of this scheme is that, similar to Free-Lagrange, it is now straightforward to advance each node in time as soon as it's neighboring nodes are identified. We facilitate this possibility with linked-list storage of the labels of nearest neighbors at each node.

The spatial nodes are built at the intersection of very fine 1-D arrays in each dimension. These n arrays, where n is the dimensionality of the problem, are lists of *possible* locations at which a node could be constructed. The product of the dimensions of these 1-D arrays give all the possible nodal positions but, other than the 1-D arrays themselves, we allocate memory only for those positions at which a node exists. We merely quantise the coordinate spatial coordinates that a node can have. Similar to AMR we only allow orthogonal dimensions so to ease the evaluation of the Laplacians with nearby boundaries. (See Embedded Curved Boundary paper in proceedings of the Numerical Simulations of Plasmas, Santa Barbara, 1998 or JCP **138**, p 585, 1997)

Adaptively is easily achieved by creating new nodes at the smaller of the light distance or the midpoint of nearest neighbors with values either zero or from interpolation, respectively. After a macro time step, nodes whose presence generated an insignificant change are deleted. The overall strategy is to aggressively generate new nodes to probe for emerging features and then to aggressively delete nodes if interesting features fail to materialize.

Time integration

Time integration is the most intricate part of this scheme. As the governing equations clearly represent an explicit time integration, the CFL condition will obviously have a controlling voice in how the integration proceeds. A typical time cycle, that works well, has the following logic:

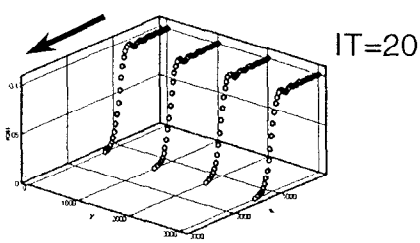
1) Starting with all the nodes at the same time level, define a new macro time level Δt and assign that time "deficit" to each node.
2) Process the a node by:
   a) Identifying all the orthogonal neighbors.
   b) If not the "youngest" node in the neighborhood, skip to the next node in the list.
   c) Find the closest neighbor, set the micro time step δt equal to the smaller of Δt and $\delta d/c$ where δd is the distance to the closest neighbor.
   d) Interpolate "older" neighbors *backwards* in time, to get the Laplacians at the current time of the center node now being advanced.
   e) Advance this node, reducing its time deficit by δt.
   f) When the node's time deficit is zero, move node to the "finished" partition of the node list.
   g) If particles with finite time deficit still exist, do the next one starting with a).
3) Delete, for economy, nodes whose presence has generated an insignificant change.
   a) Delete if a similar Laplacian could have been generated using neighbors "once removed".
   b) Delete if the functional value is near its neighborhood average
      and the function time derivative is small
      and near a small neighborhood average time derivative.

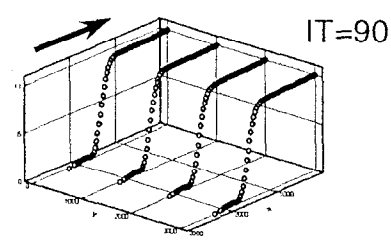
Note that step 3) enhances economy in at least 2 ways: 1) deleting nodes increases the inter-nodal distance thus allowing larger micro time steps δt (fewer "subcycles") and, obviously, 2) the fewer nodes the less work required.

Finally, when we apply this algorithm to electromagnetic pulse EMP studies, it is possible to get very steep pulses, which are too sharp to be resolved, by even this virtual mesh. In these cases, a Lax operator is applied to the update of the potentials Φ and $\vec{A}$. I replace the previous value in the update equations by some contribution from the neighborhood average if nodes are 1) close together and 2) relatively inactive.

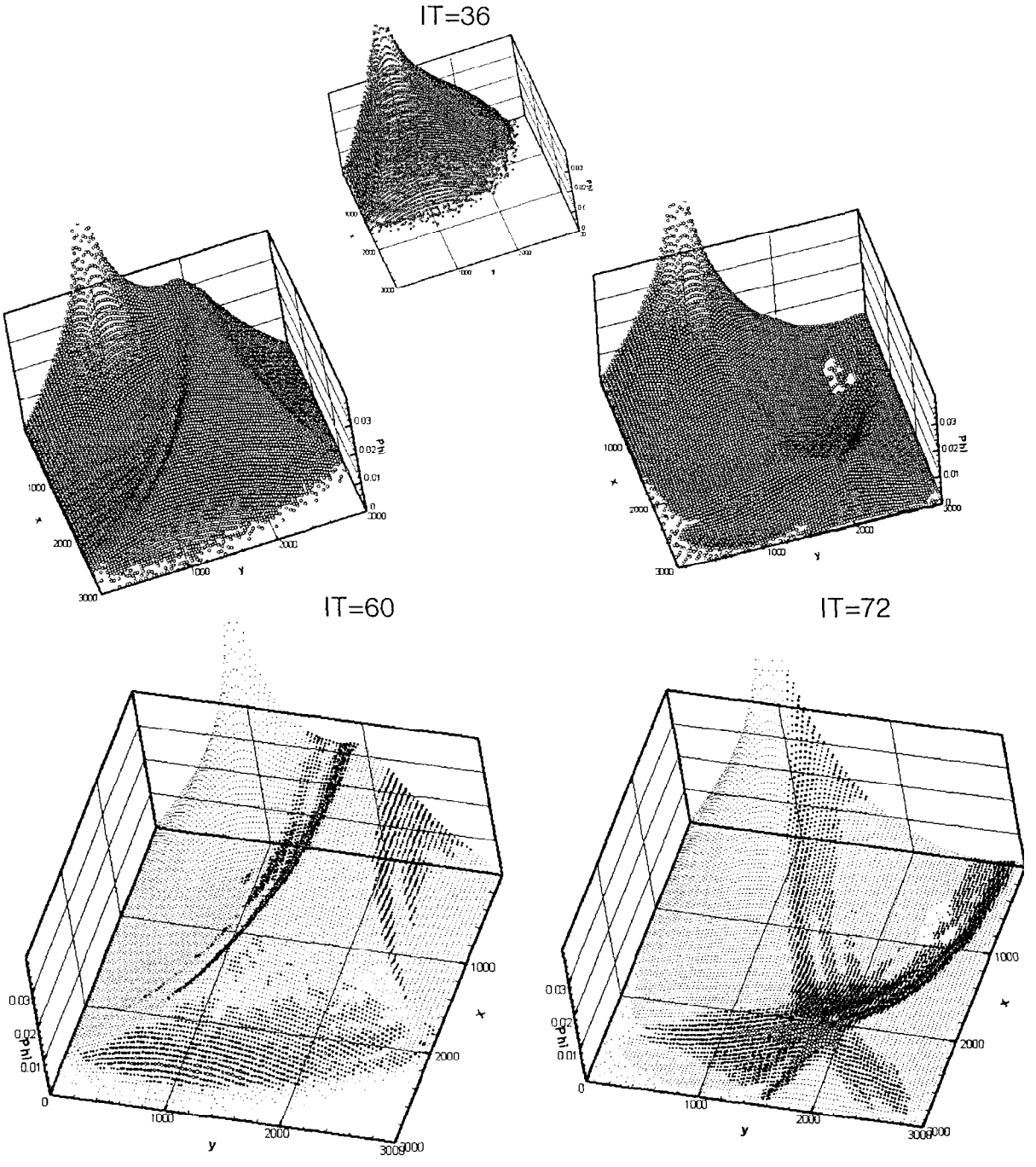
Results

This method is relatively new and has not yet been subjected to exhaustive test cases. Initial tests have concentrated on just the electrostatic potential equation and involve abruptly increasing the potential to a constant, finite value at some position in the region. In 1-D this results in a step function propagating back and forth across the box, from a zero Dirichlet on one end to the constant value on the other. Here are four non-interacting pulses rattling back and fourth in a 1-D box.





Later 2-D studies involve a similar arrangement, this time with three walls being grounded and the back wall serving as a plane of symmetry. On this back wall, I abruptly raise the potential to a finite, constant value at **one and only one** node on that wall. The resulting dynamics is that of an oscillating, growing "chocolate kiss" in which the propagation and then reflection with the grounded walls is followed by more complex interaction of the various reflected waves.



(Lower pictures have node size proportional to the time derivative.)

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.